Graph Coarsening Method for KKT Matrices Arising in Orthogonal Collocation Methods for Optimal Control Problems

Begüm Şenses*
Anil V. Rao†
University of Florida
Gainesville, FL 32611-6250
Timothy A. Davis‡
Texas A&M University
College Station, TX 77843

State-defect constraint pairing graph coarsening method is described for large sparse Karush-Kuhn-Tucker (KKT) matrices that arise from the discretization of optimal control problems via an orthogonal collocation method. The method takes advantage of the known sparsity pattern of the KKT matrix. For any particular optimal control problem, the method pairs each component of the state with its corresponding defect constraint. Following that the method forces paired rows to be adjacent in the re-ordered KKT matrix. Aggregate results are presented using a wide variety of benchmark optimal control problems and show that the method reduces the number of delayed pivots and the number of delayed pivots significantly.

I. Introduction

Efficient computation of numerical solutions to complex optimal control problems is of great interest engineering and non-engineering disciplines. In recent years, direct collocation methods have become the standard for solving optimal control problems numerically. In this paper, a continuous optimal control problem is transcribed to a nonlinear programming problem (NLP) using a previously developed Legendre-Gauss-Radau (LGR) orthogonal collocation method.[1–5] The NLP resulting from the LGR discretization can then be solved

*PhD Candidate, Department of Mechanical and Aerospace Engineering. Email: bgmsenses@ufl.edu.
†Associate Professor, Department of Mechanical and Aerospace Engineering. Associate Fellow, AIAA. E-mail: anilvrao@ufl.edu. Corresponding Author. Associate Fellow AIAA.
‡Professor, Department of Computer Science & Engineering Information & Science Engineering. E-mail: davis@tamu.edu.

using a wide variety of well established NLP solvers (for example, SNOPT[6] or IPOPT[7]). A common feature of all of the available sparse NLP solvers is that the vast majority of the computation time required to solve the NLP is the time required to solve the sparse symmetric indefinite KKT systems. Consequently, a great deal of work has been done on development of robust and efficient sparse symmetric indefinite linear solvers. Examples of well-known sparse symmetric indefinite linear solvers include MA47,[8] MA57,[9] Mumps[10] and Pardiso.[11] All of these linear solvers take advantage of the symmetry of the KKT matrix, and only some of these linear solvers take advantage of the additional sparsity because of the zero block in the KKT matrix, however, none of these linear solvers take advantage of the known sparsity pattern of the KKT matrices arising from LGR collocation method. Additionally, some of these off-the-shelf linear solvers use graph coarsening methods to partition the KKT matrices into smaller systems of approximately the same size and solve these smaller systems in parallel, however, none of these linear solvers use graph coarsening method to decrease the number of delayed pivots.

In this paper we propose a state-defect constraint pairing graph coarsening method tailored for large-scale optimal control problems using LGR orthogonal collocation method. The goal of the graph coarsening method is to design a more computationally efficient and more robust numerical factorization of the sparse indefinite KKT matrices by decreasing the number of delayed pivots and preventing the insufficient memory allocation. The graph coarsening method is performed using only the properties of the continuous optimal control problem, the number of collocation points in each mesh iteration, and the known sparsity pattern of the KKT matrices resulting from the LGR orthogonal collocation method. The method is applied to a wide variety of benchmark optimal control problems where the benefits of the graph coarsening method are demonstrated.

II. Solution of Optimal Control Problems

An optimal control problem determines the state, $\mathbf{y}(\tau) \in \mathbb{R}^n$, and the control $\mathbf{u}(\tau) \in \mathbb{R}^m$, that minimize a cost functional subject to dynamic, path and event constraints. Throughout this research optimal control problems are transcribed into nonlinear programming problems of the form

$$\min \quad f(\mathbf{x}) \tag{1}$$

$$\text{s.t.} \quad \mathbf{c}(\mathbf{x}) = \mathbf{0} \tag{2}$$

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \tag{3}$$

by a general purpose MATLAB software GPOPS-II[12] that employs Legendre-Gauss-Radau (LGR) orthogonal collocation method. A detailed explanation on LGR orthogonal collocation methods can be found in Refs. 1–5. The NLP of Eqs. (1) - (3) is solved using the open-source NLP solver IPOPT[7] in full Newton (second derivative) mode. IPOPT performs an interior-point method with a filter-based line search. The details of this method are beyond the scope of this research but can be found in Ref. 13. IPOPT solves a sparse symmetric KKT system at each iteration where the KKT matrix has the form

$$\begin{bmatrix} \mathbf{H}^{(k)} & \bigtriangledown \mathbf{c}(\mathbf{x}^{(k)})^{\mathsf{T}} \\ \bigtriangledown \mathbf{c}(\mathbf{x}^{(k)}) & 0 \end{bmatrix}. \tag{4}$$

In Eq. (4) $\mathbf{H}^{(k)}$ is

$$\mathbf{H}^{(k)} = \bigtriangledown^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^{(k)}) + (\underline{\mathbf{X}}^k)^{-1}\underline{\mathbf{D}}^k + (\overline{\mathbf{X}}^k)^{-1}\overline{\mathbf{D}}^k. \tag{5}$$

In Eq. (5), $\mathcal{L}$ is the Lagrangian function, $\overline{\mathbf{X}} = diag(\mathbf{x}_{\max} - \mathbf{x})$, $\underline{\mathbf{X}} = diag(\mathbf{x}_{\min} - \mathbf{x})$, $\overline{\mathbf{D}}^k = diag(\overline{\mathbf{d}}^k)$ and $\underline{\mathbf{D}}^k = diag(\underline{\mathbf{d}}^k)$. If μ is the Lagrange multiplier that corresponds to the simple bounds on $\mathbf{x}$, $\underline{\mathbf{d}}$ and $\overline{\mathbf{d}}$ are defined as follows.

$$\overline{\mathbf{d}} = \mu[\overline{\mathbf{X}}]^{-1}\mathbf{e}$$
$$\underline{\mathbf{d}} = \mu[\underline{\mathbf{X}}]^{-1}\mathbf{e}$$

where $\mathbf{e} = (1, ..., 1)^\mathsf{T}$. Sparse symmetric indefinite KKT matrices arising at every iteration of IPOPT is solved using MA57.[9]

MA57 solves the KKT matrices in three phases: analysis phase, numerical factorization phase, and the solution phase. While a detailed description of the methods used in MA57 is beyond the scope of this paper (see Refs. 14–16), below MA57 analysis and numerical factorization phases are briefly described. In the analysis phase, MA57 permutes the KKT matrix using a fill-reducing algorithm, determines the elimination tree structure, postorders the elimination tree and forecasts the storage needs for the numerical factorization phase. In the numerical factorization phase MA57 employs a multifrontal method that uses either a 1×1 or 2×2 pivot to find the LDL^T factorization of the permuted KKT matrix. Solution accuracy of an indefinite matrix depends on the numerical values of the pivot. Therefore, MA57 examines the numerical stability of pivots via a pivot threshold test is given as follows. Accept the diagonal element of the k^{th} row, $a_{k,k}$, as the 1×1 pivot if

$$|a^{-1}_{k,k}|\alpha_k \leq u^{-1} \tag{6}$$

is satisfied or accept $\mathbf{B}_{k,k}$ as the 2×2 pivot if

$$|\mathbf{B}^{-1}_{k,k}| \begin{bmatrix} \beta_k \\ \beta_{k+i} \end{bmatrix} \leq \begin{bmatrix} u^{-1} \\ u^{-1} \end{bmatrix}, \quad i > 1 \tag{7}$$

is satisfied ($0 < u < 0.5$). α_k, β_k and $\mathbf{B}_{k,k}$ are given as follows.

$$\alpha_k = \max_{j \geq k+1} |a_{k,j}| \tag{8}$$

$$\beta_k = \max_{j \geq k+2} |a_{k,j}| \tag{9}$$

and

$$\mathbf{B}_{k,k} = \begin{bmatrix} a_{k,k} & a_{k,k+i} \\ a_{k+i,k} & a_{k+i,k+i} \end{bmatrix}, \quad i > 1 \tag{10}$$

If a pivot cannot satisfy neither the 1×1 nor the 2×2 pivot criterion then that pivot row is delayed to be eliminated later in the factorization phase and the pivot search moves on to the next diagonal. When MA57 analysis phase predicts the storage needs of the numerical factorization phase, it does not take into account the delayed pivots. Delayed pivots increase the fill-in (number of nonzeros in the $\mathbf{L}$ matrix) and the storage needs of the factorization phase. Therefore, if there is a large number of delayed pivots, MA57 may fail due to insufficient memory allocation. In that case, MA57 stops the numerical factorization phase, reallocates a larger memory for the numerical factorization phase.

III. KKT Matrix Properties & Observations on Delayed Pivots

One of the most important properties of the KKT matrices arising from LGR collocation is their known sparse structure. Figure 1 shows an example of the sparsity pattern of a KKT matrix arising from LGR collocation. As seen from Fig. 1, the Lagrangian Hessian and the constraint Jacobian matrices have a great deal of structure. The Lagrangian Hessian and the off-diagonal part of the constraint Jacobian matrices consist of purely diagonal blocks.

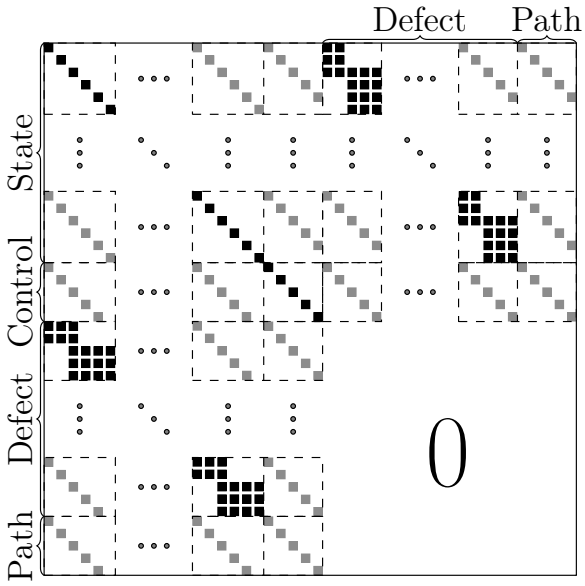


Figure 1. Sparsity pattern of KKT matrices arising from LGR collocation

The KKT matrix given in Fig. 1 is a symmetric indefinite matrix which is factorized using 1×1 and 2×2 pivots. The 1×1 pivot properties of the KKT matrix shown in Fig. 1 are given as follows: the state and control rows have always nonzero diagonals therefore these rows may satisfy the 1×1 pivot criterion given in Eq. (6), on the other hand, the defect and path constraint rows have always zero diagonals therefore these rows cannot satisfy the 1×1 pivot criterion unless the fill-in caused by the state and/or control rows is large enough. Following the 1×1 pivot properties, the 2×2 pivot properties of the KKT matrix shown in Fig. 1 are given. Constraints cannot form a nonsingular 2×2 pivot with other constraints because the 2×2 pivot formed by two constraints consists of only zeros, on the other hand, constraints may form a 2×2 pivot of the form

$$\begin{pmatrix} \times & \times \\ \times & 0 \end{pmatrix} \tag{11}$$

with variables.

Another important property of the KKT matrices arising from LGR collocation is that the number of states are equal to the number of defect constraints.

Properties of KKT matrices arising from LGR collocation are described. We now give our observations on delayed pivots that occur during the numerical factorization of the KKT matrices arising from LGR discretization of continuous optimal control problems. Our first observation is about the relationship between the number of floating point operations in the

numerical factorization phase and the number of delaye pivots. As seen in Fig. 2 the number of floating point operations in the numerical factorization phase increases with the number of delayed pivots. Our second observation is about the nature of the delayed pivots. Most

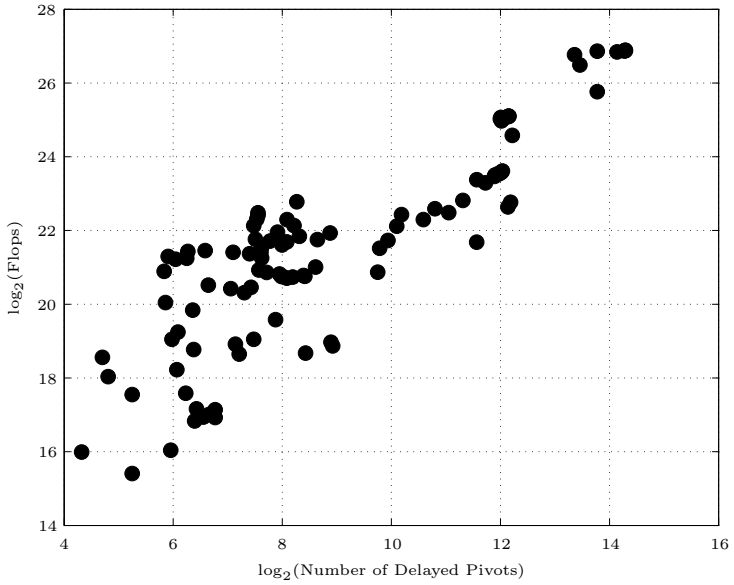


Figure 2. $\log_2$**(Number of Delayed Pivots) vs** $\log_2$**(Number of Floating Point Operations)**

of the delayed pivots belong to the constraint Jacobian part of the KKT matrices and they correspond to defect and path constraints of the discretized optimal control problem.

IV. Graph Coarsening Method

The state-defect constraint pairing graph coarsening method uses a matching based graph coarsening method. The difference between the matching based graph coarsening methods in the literature[17–19] and the proposed method is the choice of matching pairs. The matching based graph coarsening methods in the literature use the characteristics of the original graph to decide the pairs of vertices. On the other hand, the pairs of the state-defect constraint pairing graph coarsening method is a priori known as a result of the known sparsity pattern of the KKT matrices arising from LGR collocation method.

In this section, we concentrate on reducing the number of delayed pivots that correspond to defect constraints of a continuous optimal control problem by performing a graph coarsening method tailored for KKT matrices arising from LGR collocation. As already mentioned, defect constraints have zero diagonals. Therefore those rows in the KKT matrix cannot satisfy the 1×1 pivot criterion unless the fill-in caused by a state or control component is large enough. Moreover, a defect constraint may form a 2×2 pivot with a state or a control component if that defect constraint is a function of that state or control component. Additionally, most of the delayed defect constraint pivots belong to a frontal matrix with one fully summed row and column. All of these properties of defect constraints and observations on delayed defect constraints lead to the conclusion that forcing a state or control pivot to be in the same frontal matrix with a defect constraint may decrease the possibility of this defect constraint getting delayed. Consequently, the following questions arise: First, which is bet-

ter, state or control component pair with a defect constraint? Second, which state or control component decreases the possibility of defect constraints getting delayed? State components are "better" pairs for defect constraints as compared with the control because the number of defect constraints are equal to the number of state components so there is a state pair for each defect constraint independent of the continuous optimal control problem. On the other hand, the number of control components may or may not be equal to the number of defect constraints depending on the continuous optimal control problem. If the number of control components is less than the number of defect constraints, some of the defect constraints remain without a pair. All together, if a defect constraint and its corresponding state are in the same frontal matrix, that defect constraint is less likely to be delayed.

This study proposes a graph coarsening method that forces state-defect constraint pairs to be adjacent in the reordered KKT matrix. Although two consecutive rows are not guaranteed to be in the same frontal matrix during the numerical factorization phase, forcing state-defect constraint pairs to be consecutive increases the possibility of them being in the same frontal matrix. The first step of the state-defect constraint pairing graph coarsening method is finding the state-defect constraint pairs. The state-defect constraint pairs are found using the properties of the continuous time optimal control problem, the number of collocation points, and the known sparsity pattern of the KKT matrix. The next step of the graph coarsening method is matching (collapsing) the vertices of the state-defect constraint pairs and representing each pair with a multinode in the coarsened graph. The multinodes consist of the union of vertices adjacent to the state - defect constraint pair nodes. After coarsening, a fill-reducing ordering is performed on the coarsened matrix and finally the reordered and coarsened matrix is uncoarsened. As a result of these steps, state-defect constraint pairs at each collocation point are kept adjacent in the reordered KKT matrix. In Section A it is demonstrated that the state-defect constraint pairing graph coarsening method decreases both the number of delayed pivots and therefore flop counts during the numerical factorization phase.

A. Results

The state-defect constraint pairing graph coarsening method was performed on large sparse KKT matrices. The KKT matrices associates with the nonlinear programming problems that arise from LGR collocation of continuous optimal control problems of varying complexity taken from Refs. 20–28. The steps after performing the graph coarsening method was solving KKT matrices by an off-the-shelf linear solver MA57 and comparing the performance of MA57 numerical factorization phase with and without the graph coarsening method. Before giving a detailed discussion on MA57 numerical factorization phase performance, we introduce some terminology that is used throughout the performance analysis. Firstly, we refer the KKT matrices that are solved without a graph coarsening method as the original matrices and the KKT matrices that are solved after performing a graph coarsening method as the coarsened matrices. Moreover, MA57 returns two $\mathbf{L}$ matrices; one after the symbolic factorization phase and one after the numerical factorization phase. As aforementioned, those two $\mathbf{L}$ matrices may be different for indefinite matrices because the symbolic analysis phase does not take into account the 2×2 pivots and delayed pivots. We refer the $\mathbf{L}$ matrix arising from the symbolic factorization phase as the expected $\mathbf{L}$ matrix and the $\mathbf{L}$ matrix arising from the numerical factorization phase as the actual $\mathbf{L}$ matrix. We also use

terminology such as the expected fraction and the actual fraction. In this study, fraction refers to the ratio of the state-defect constraint pairs that are in the same frontal matrix to the total number of state-defect constraint pairs. Additionally, the expected and the actual fraction indicate the fraction in the expected and the actual **L** matrices respectively. After introducing the terminology, now we can start the performance analysis. Figure 3 compares the expected **L** matrices arising from the symbolic factorization of original matrices with the actual **L** matrices arising from the numerical factorization of original matrices in terms of the state-defect constraint pair fraction. Figure 3(a) illustrates that the state-defect constraint pair fraction in the expected **L** matrices changes between 0 to 0.3. As it is observed from Figs. 3(b) and 3(c) state-defect constraint pair fraction in the actual **L** matrices are higher than the state-defect constraint pair fraction in the expected **L** matrices. This shows us that the state-defect pairs tend to be in the same frontal matrix in the actual **L** matrices. If the state and its corresponding defect constraint are not in the same frontal matrix after the symbolic factorization phase, defect constraints get delayed and end up in the same frontal matrix with their corresponding state. The state-defect constraint pairing graph coarsening method aims to decrease the number of delayed pivots by increasing the state-defect constraint fraction in the expected **L** matrices. Figure 4 shows the state-defect constraint pair fraction in the expected **L** matrix. As seen from Fig. 4 performing the graph coarsening method increases the state-defect constraint pair fraction drastically. Figure 5 shows the ratio of the floating point operations in the numerical factorization of coarsened matrices to the floating point operations in the numerical factorization of original matrices. It is observed from Fig. 5 that if an original matrix has approximately more than 1000 delayed pivots, then factorizing the coarsened matrix always decreases the number of floating point operations. On the other hand, if the number of delayed pivots of the original matrix is less than 1000, factorizing the original matrix may be more efficient than factorizing the coarsened matrix.

V. Conclusions

First, a state-defect constraint pairing graph coarsening method has been described for Karush-Kuhn-Tucker linear systems associated with the nonlinear programming problems that arise from the discretization of continuous optimal control problems using the Legendre-Gauss-Radau collocation method. It was found that if the number of delayed pivots in the original matrix is large enough, the graph coarsening method decreases the number of delayed pivots and floating point operations.

Acknowledgments

The authors gratefully acknowledge support for this research from the the U.S. Space and Warfare Systems Command under Grant N65236-13-1-1000.

References

[1]Garg, D., Patterson, M. A., Darby, C. L., Francolin, C., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Opti-

mal Control Problems via a Radau Pseudospectral Method," *Computational Optimization and Applications*, Vol. 49, No. 2, June 2011, pp. 335–358. DOI: 10.1007/s10589–00–09291–0.

[2]Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, Vol. 46, No. 11, November 2010, pp. 1843–1851. DOI: 10.1016/j.automatica.2010.06.048.

[3]Garg, D., Hager, W. W., and Rao, A. V., "Pseudospectral Methods for Solving Infinite-Horizon Optimal Control Problems," *Automatica*, Vol. 47, No. 4, April 2011, pp. 829–837. DOI: 10.1016/j.automatica.2011.01.085.

[4]Darby, C. L., Hager, W. W., and Rao, A. V., "An *hp*–Adaptive Pseudospectral Method for Solving Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 32, No. 4, July–August 2011, pp. 476–502.

[5]Darby, C. L., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization Using a Variable Low-Order Adaptive Pseudospectral Method," *Journal of Spacecraft and Rockets*, Vol. 48, No. 3, May–June 2011, pp. 433–445.

[6]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM journal on optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[7]Biegler, L. T. and Zavala, V. M., "Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, Vol. 33, No. 3, 2009, pp. 575–582.

[8]Duff, I. S. and Reid, J. K., "MA47, a Fortran code for direct solution of indefinite sparse symmetric linear systems," *Report RAL*, 1995, pp. 95–001.

[9]Duff, I. S., "MA57—a code for the solution of sparse symmetric definite and indefinite systems," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 30, No. 2, 2004, pp. 118–144.

[10]Amestoy, P. R., Duff, I. S., LâĂŹExcellent, J.-Y., and Koster, J., *MUMPS: a general purpose distributed memory sparse solver*, Springer, 2001.

[11]Schenk, O. and Gärtner, K., "Two-level dynamic scheduling in PARDISO: Improved scalability on shared memory multiprocessing systems," *Parallel Computing*, Vol. 28, No. 2, 2002, pp. 187–197.

[12]Patterson, M. A. and Rao, A. V., "GPOPS- II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp–Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 39, No. 3, 2013.

[13]Wächter, A. and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, Vol. 106, No. 1, 2006, pp. 25–57.

[14]Duff, I. S. and Reid, J. K., *MA27-a set of Fortran subroutines for solving sparse symmetric sets of linear equations*, UKAEA Atomic Energy Research Establishment, 1982.

[15]Duff, I. S. and Reid, J. K., "The multifrontal solution of indefinite sparse symmetric linear," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 9, No. 3, 1983, pp. 302–325.

[16]Liu, J. W., "The multifrontal method for sparse matrix solution: Theory and practice," *SIAM review*, Vol. 34, No. 1, 1992, pp. 82–109.

[17]Karypis, G. and Kumar, V., "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on scientific Computing*, Vol. 20, No. 1, 1998, pp. 359–392.

[18]Bui, T. N. and Jones, C., "A Heuristic for Reducing Fill-In in Sparse Matrix Factorization." *PPSC*, 1993, pp. 445–452.

[19]Hendrickson, B. and Leland, R. W., "A Multi-Level Algorithm For Partitioning Graphs." *SC*, Vol. 95, 1995, pp. 28.

[20]Zhao, Y. J., "Optimal patterns of glider dynamic soaring," *Optimal control applications and methods*, Vol. 25, No. 2, 2004, pp. 67–89.

[21]Ledzewicz, U. and Schättler, H., "Analysis of optimal controls for a mathematical model of tumour anti-angiogenesis," *Optimal Control Applications and Methods*, Vol. 29, No. 1, 2008, pp. 41–57.

[22]Bryson, A. E., *Applied optimal control: optimization, estimation and control*, CRC Press, 1975.

[23]Şenses, B. and Rao, A. V., "Optimal Finite-Thrust Small Spacecraft Aeroassisted Orbital Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 6, 2013, pp. 1802–1810.

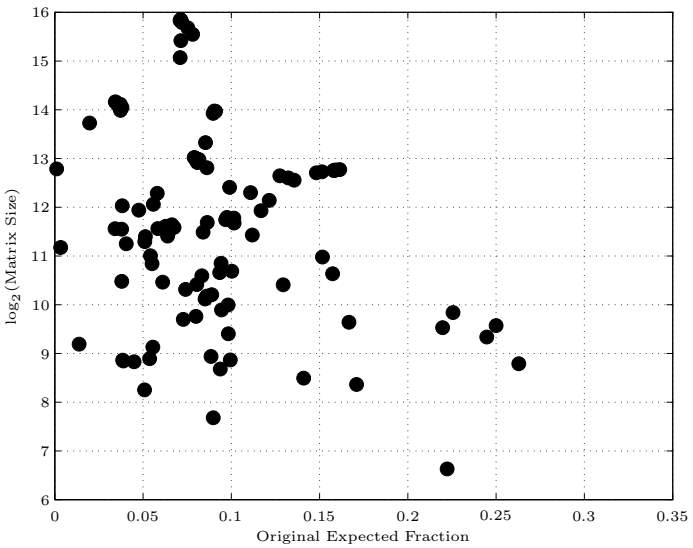
[24]Bulirsch, D. R., Nerz, D. M. E., Pesch, P.-D. D. H. J., and von Stryk, D. M. O., *Combining direct and indirect methods in optimal control: Range maximization of a hang glider*, Springer, 1993.

[25]Sakawa, Y., "Trajectory planning of a free-flying robot by using the optimal control," *Optimal Control Applications and Methods*, Vol. 20, No. 5, 1999, pp. 235–248.

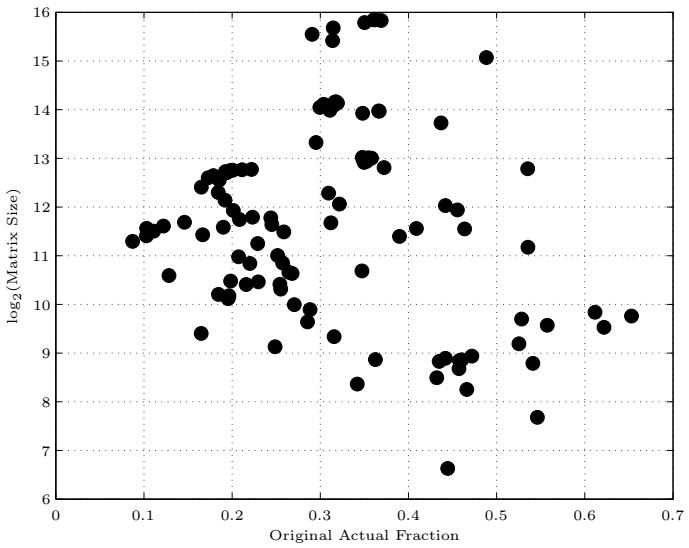
[26]Schubert, K. F. and Rao, A. V., "Minimum-Time Low-Earth Orbit to High-Earth Orbit Low-Thrust Trajectory Optimization," *2013 AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, South Carolina, 12 - 15 August 2013, pp. AAS Paper 13–926.

[27]Goddard, R. H., "A Method of Reaching Extreme Altitudes." *Nature*, Vol. 105, 1920, pp. 809–811.

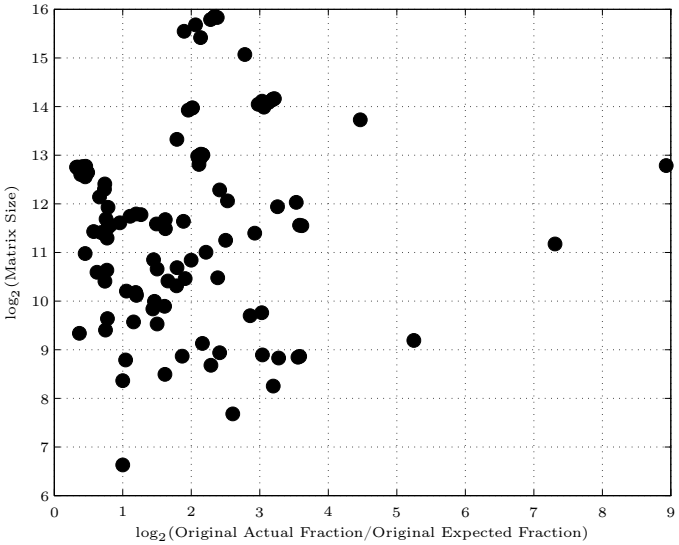
[28]Betts, J. T., *Practical methods for optimal control and estimation using nonlinear programming*, Vol. 19, Siam, 2010.



(a) $\log_2$(Matrix Size) vs. Original Expected Fraction



(b) $\log_2$(Matrix Size) vs. Original Actual Fraction



(c) $\log_2$(Matrix Size) vs. $\log_2\left(\frac{\text{Original Actual Fraction}}{\text{Original Expected Fraction}}\right)$

Figure 3. Comparison of the expected L matrices arising from the symbolic factorization of original matrices with the actual L matrices arising from the numerical factorization of original matrices in terms of state-defect constraint pair fraction vs. $\log_2$**(Matrix Size)**

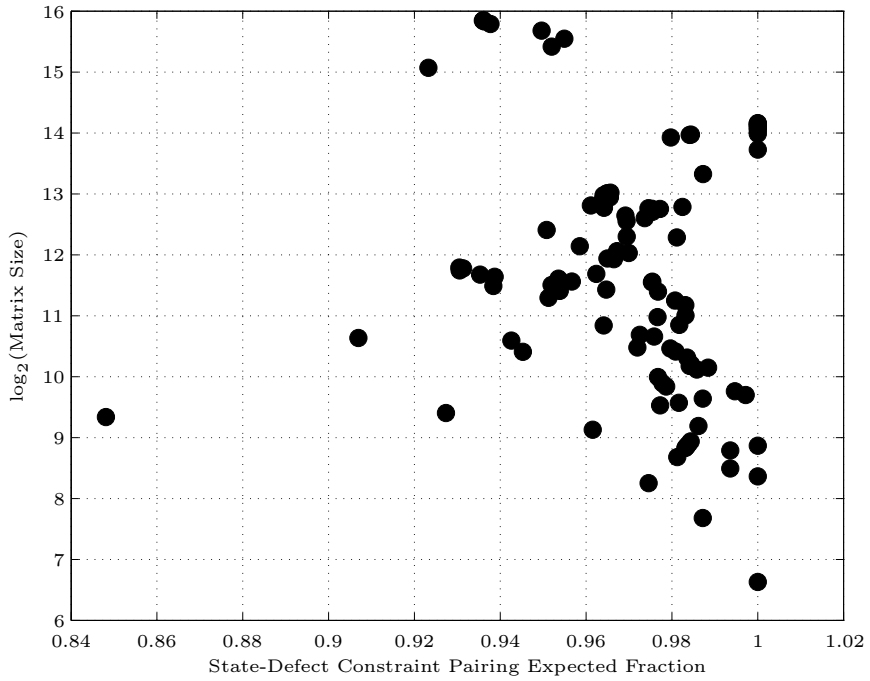


Figure 4. The expected state-defect constraint pair fraction in coarsened matrices vs. $\log_2$(Matrix Size)

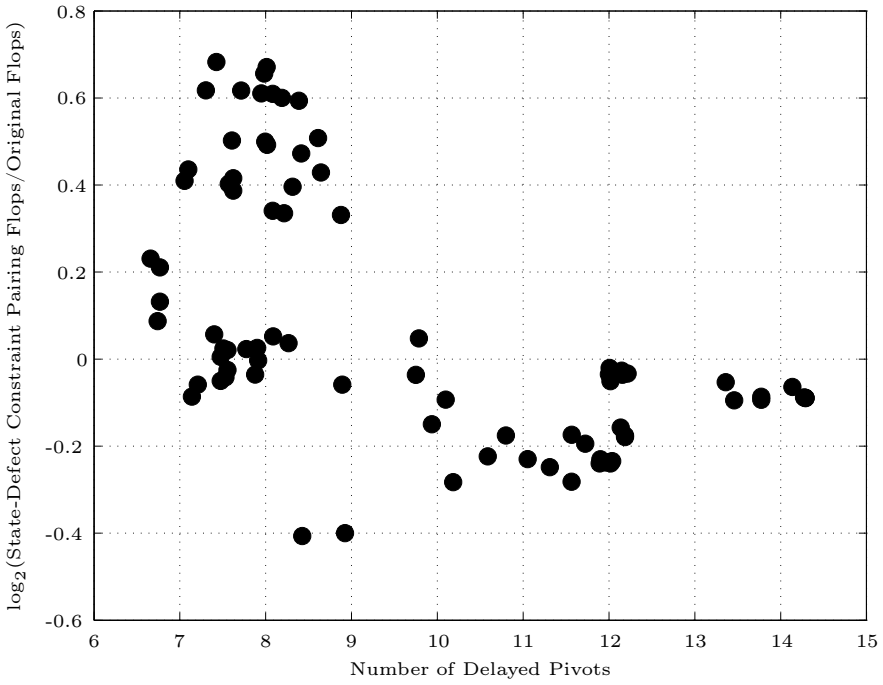


Figure 5. Number of delayed pivots in original matrices vs. fraction of floating point operations with and without state-defect constraint pairing